

NASA
IN-10-CR
293534
P-46

A Design Optimization Process for Space Station Freedom

Robert G. Chamberlain
George Fox
William H. Duquette

June 15, 1990



National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

(NASA-CR-186851) A DESIGN OPTIMIZATION
PROCESS FOR SPACE STATION FREEDOM (JPL)

46 p

CSCL 22B

N90-25175

Unclass

G3/18 0293534

2-1

1. Report No. JPL Publication 90-23		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle A DESIGN OPTIMIZATION PROCESS FOR SPACE STATION FREEDOM				5. Report Date June 15, 1990	
				6. Performing Organization Code	
7. Author(s) Robert G. Chamberlain, George Fox, William H. Duquette				8. Performing Organization Report No.	
9. Performing Organization Name and Address JET PROPULSION LABORATORY California Institute of Technology 4800 Oak Grove Drive Pasadena, California 91109				10. Work Unit No.	
				11. Contract or Grant No. NAS7-918	
				13. Type of Report and Period Covered	
12. Sponsoring Agency Name and Address NATIONAL AERONAUTICS AND SPACE ADMINISTRATION Washington, D.C. 20546				14. Sponsoring Agency Code RE241 BH-476-30-00-00-00	
15. Supplementary Notes					
16. Abstract <p>We are working with the Space Station Freedom Program to develop and implement a process for design optimization. Because the relative worth of arbitrary design concepts cannot be assessed directly, comparisons must be based on designs that provide the same performance from the point of view of station users; such designs can be compared in terms of life-cycle cost. Since the technology required to produce a space station is widely dispersed, a decentralized optimization process is essential. This publication provides a formulation of the optimization process and describes the mathematical models designed to facilitate its implementation.</p>					
17. Key Words (Selected by Author(s)) LIFE-CYCLE COSTING SYSTEMS ANALYSIS NONLINEAR PROGRAMMING APPLICATIONS SPACE STATION				18. Distribution Statement UNCLASSIFIED -- UNLIMITED	
19. Security Classif. (of this report) UNCLASSIFIED		20. Security Classif. (of this page) UNCLASSIFIED		21. No. of Pages 48	
				22. Price	



A Design Optimization Process for Space Station Freedom

Robert G. Chamberlain
George Fox
William H. Duquette

June 15, 1990



National Aeronautics and
Space Administration

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California

The research described in this publication was carried out by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not constitute or imply its endorsement by the United States Government or the Jet Propulsion Laboratory, California Institute of Technology.

ABSTRACT

We are working with the Space Station Freedom Program to develop and implement a process for design optimization. Because the relative worth of arbitrary design concepts cannot be assessed directly, comparisons must be based on designs that provide the same performance from the point of view of station users; such designs can be compared in terms of life-cycle cost.

Since the technology required to produce a space station is widely dispersed, a decentralized optimization process is essential. This publication provides a formulation of the optimization process and describes the mathematical models designed to facilitate its implementation.

FOREWORD

The primary purpose of this publication is to provide archival documentation of the fundamental analyses on which the System Design Tradeoff Model is based. It is our hope that we have been sufficiently thorough to permit replication and extension of the analyses, sufficiently complete to permit assessment of the validity and usefulness of the approach, and sufficiently clear to facilitate understanding by interested readers.

While the derivation of the model is fully presented, it is intentional that the emphasis of the publication is on the process that the model is designed to support.

SUMMARY

The Space Station Freedom Program is designing and building a manned space station that will be assembled in orbit and operated for thirty years or more.

The designers are, of course, seeking the best design – but the term "best" must be interpreted in a broad context. Many aspects of the "best" design require policy decisions that must be made at much higher levels of management than those that deal with the details of design.

A formal statement of the mathematical design optimization problem is a nonlinear program: find the sizes of subsystems and the values of design parameters that minimize life-cycle cost subject to constraining performance specifications. Careful examination of the problem reveals that the lowest life-cycle cost is obtained by the design that meets all of the performance specifications with no slack. To reach that conclusion, it must be assumed that larger subsystems cost more and consume more on-board resources than smaller versions of the same subsystems.

A central authority, however, cannot – and should not – solve this problem in its entirety, because much of the necessary information is geographically, organizationally, and temporally dispersed. The technique of Lagrangian relaxation is used to decompose the problem into a system-level design optimization problem and a collection of subsystem-level problems. A decentralized process for alternately solving these problems is described.

The System Design Tradeoff Model computer program, SDTM, which was developed to help solve the system-level problem and to facilitate the operation of the decentralized process, is described. An appendix details the algorithms it uses.

Brief descriptions of extensions to the analysis conclude this publication: the sufficient conditions for the easy solution to the nonlinear program are replaced by necessary conditions. The level of detail at which the station is described is discussed; procedures for aggregating or resolving the description are presented. Determination of the optimal growth in capacities with the passage of time is discussed. A reformulation of the problem to deal with uncertainties is presented.

ACKNOWLEDGMENTS

Dr. Jeffrey L. Smith has inspired and managed the development of this work since its inception. He has championed the use of user requirements specifications and life-cycle costs in Space Station Freedom decision-making so successfully that it is realistic to hope for widespread use of the process described here.

Dr. Orin H. Merrill and Dr. James W. Doane, at the Space Station Freedom Program Office in Reston, Virginia, have been instrumental in application of the process so far, as well as providing the funding for the development activity. Dr. Merrill, in particular, with the assistance of Anita Adams, has played a key role in the construction of a usable description of the Space Station Freedom design space.

Many others have contributed significantly, in some cases for several years, to the development of the process, to the creation of the SDTM computer program, to the description of the Space Station Freedom design space, and to general support of the activity. Among these are Jim Akkerman, Dave Bates, Chet Borden, Bob Brodowski, Barry Brown, Govind Deshpande, Larry DiLullo, Bob Easter, Don Ebbeler, Don Gantzer, Bill Gray, Ann Griesel, Hamid Habib-agahi, Tony Hagar, Robert Hall, Byron Jackson, Ed Jorgensen, Frank Judnick, Tim Meier, Art Metz, Ralph Miles, Charles Nainan, Gary Oleson, Mark Olson, Lori Paul, Dave Porter, Tony Rice, Larry Seeley, Bob Shishko, Laura Steele, Dan Urbina, Erik Wenberg, and Greg Williams.

CONTENTS

1	THE SSFP DESIGN OPTIMIZATION PROBLEM.....	1
2	SOLUTION OF THE MATHEMATICAL PROBLEM.....	5
3	DECENTRALIZATION OF THE PROCESS.....	11
4	DECOMPOSITION	13
4.1	RESOURCE SIZES	13
4.2	LAGRANGE MULTIPLIERS (IMPLICIT PRICES)	14
4.3	SYSTEM-LEVEL DESIGN PARAMETERS	14
4.4	SUBSYSTEM-LEVEL DESIGN PARAMETERS	15
4.5	DEFINITION OF SUBSYSTEM LIFE-CYCLE COST	16
5	THE SUBSYSTEM-LEVEL DESIGN OPTIMIZATION PROBLEM	17
6	THE SYSTEM-LEVEL DESIGN OPTIMIZATION PROBLEM	19
6.1	SIZING AND (IMPLICIT) PRICING	19
6.2	COSTING.....	20
7	EXTENSIONS	21
7.1	NON-NEGATIVITY OF IMPLICIT PRICES	21
7.2	AGGREGATION AND RESOLUTION OF RESOURCES.....	21
7.3	DESIGN TRAJECTORIES AND OPTIMAL GROWTH.....	22
7.4	UNCERTAINTIES	23

APPENDICES

A.	ALGORITHMIC DESCRIPTION OF THE SDTM COMPUTER PROGRAM	27
B.	THE EXPRESSION OF LEVELIZED AND LIFE-CYCLE COSTS	39

Figures

1.	The SSFP Design Optimization Problem.....	3
2.	Design Space for the Space Station Freedom Logo.....	6
3.	The <i>power, reboost</i> Slice of the SSF Design Space.....	7
4.	How Big is Big Enough?.....	8
5.	Solution of the Nonlinear Program	9
6.	New Sizes for New Technology.....	10
7.	The Design Optimization Problem for Subsystem <i>i</i>	16

SECTION 1

THE SSFP DESIGN OPTIMIZATION PROBLEM

The design problem facing the Space Station Freedom Program (SSFP) can be stated as follows:

Select design concepts and parameters
for a manned space station that will do great things
within obtainable funding.

The purpose of this publication is to describe a process for solving this problem. Part of the process is political, part technical. Those parts must be distinguished, then the technical part of the process can be analyzed in depth.

First, let us extract the selection of design concepts, the definition of "great things," and the negotiation of funding as policy issues to be dealt with by the highest level of SSFP management, in conjunction with other NASA offices, the U.S. Office of Management and Budget, the U.S. Congress, and the international partners in Space Station Freedom.

The amounts of funds available and the annual and organizational distributions of those funds are determined by the political process. Within those budgetary constraints, the funds should be used as effectively as possible. Hence, total cost should be minimized. The life-cycle cost, defined as the sum of the present values of all of the costs incurred over the lifetime of the station, provides a suitable singular measure of total cost.

Even after the fundamental character of the station has been chosen, it is impossible to quantify the relative values (that is, worths) of stations that do different things. To compare design alternatives, then, they must be placed in stations that have been made indistinguishable from the point of view of performance; they can then be compared in terms of cost. Once a design choice has been made on this basis, the choice can be expected to be valid for all stations that are sufficiently similar to the chosen station. The robustness of this conclusion depends primarily on the maturity of the station design, particularly on the maturity of those parts that depend on or affect the studied design alternatives.

Several additional policy issues must be dealt with by engineering management before the comparison of design alternatives can be approached mathematically: astronaut safety requirements, reliability standards, schedules, and congressional mandates. Policy decisions on these issues will be used as screens, so that only those design choices that pass them will enter into the mathematical optimization process.

The technical portion of the design problem can be stated as follows:

For a given manned space station design concept,
select design alternatives (from those that pass certain screens)
to minimize life-cycle cost
while meeting performance specifications.

We model the space station as a system, the performance of which must be quantified before it can be specified. This is accomplished by identifying important resources to be provided, such as power and working space, and then measuring performance by the amounts of these resources produced by the station. The space station system is then divided into subsystems, each of which produces a single resource and consumes other resources. One of the

consequences of this modeling structure is that costs associated with a subsystem can be readily attributed to the resource produced. We have, in fact, named the subsystems by the resources they produce and have sometimes used the term *resource* as shorthand for the phrase "resource-producing subsystem."

Whether one should specify the amounts of resources produced or the amounts made available to the scientific payloads that will be users of the station depends on the situation. If a user-amount perspective is taken, the amounts to be produced must be computed by adding the station's self-consumption of resources to the specified user amounts. A potential problem with an approach based on this perspective is that some resultant subsystem sizes may be too far from those for which designs have been studied to be realizable without major new design studies. As time progresses, design decisions and contractual commitments further reduce the set of sizes that is realistically available.

If a fixed-size perspective is taken, in which one specifies (*fixes*) the amounts of resources produced (that is, the *sizes* of the subsystems), the amounts available for station users must be computed as residuals after accounting for the station's self-consumption. There may be "shortfalls" of some resources – smaller amounts than are needed by users or even smaller amounts of some resources than are needed to operate the station. In any case, alternative designs would have to be compared in terms of many numbers: the amounts of each resource available to users, in addition to the life-cycle cost.

Each of these perspectives has its appropriate time in the process. First, the fixed-size perspective, using a baselined design, is taken to determine the user amounts that would be left. A political process is then invoked to establish what user amounts (and what budgets) are acceptable. The established amounts are then baselined.

The baselined user amounts are then used in a design optimization process to compare technological alternatives. Technology is assumed to be continuously, rather than discretely variable, with costs, production, and consumption interpolated smoothly between known point designs that are used to describe the space of possible designs.

Now and then, the design space should be redescribed by again taking the fixed-size perspective and re-baselining the design, using realizable sizes that are close to those implied by the user-amount specifications. The user amounts (and budgets) that result from this design can then be compared to assess alternative baseline designs. These comparisons are inherently more difficult than those made with fixed user amounts, because many factors must be compared and the preferences of many different factions must be taken into account.

These two viewpoints should be taken alternately until the set of user amounts of resources implied by a set of realizable sizes is satisfactory. Then, the resultant design should be built.

Since the construction and operation of the station will take place over an extended period of time, this process can – and should – be continued throughout the lifetime of the station. The formulas and data that describe cost, production, and consumption must be updated to reflect the effects of decisions that have been made. For example, once a contract has been let to construct a closed-loop life-support system, the costs associated with an open-loop system must include an estimate of the costs to end the closed-loop contract.

Returning to the topic of specifying performance, consider that the production and use of station resources, like the incurring of costs, occurs over the lifetime of the station. The funding agencies' current preferences for funds to be expended at other times than the present defines the discount rate that is used in the computation of present values. If we assume that the same

discount rate also captures the station users' current preference for resources to be received at various times, then we can aver that discounted present values of the amounts of resources supplied to users can be meaningfully compared. (This is equivalent to assuming that the users' best current estimate of the value of an additional unit of a resource varies with the time of delivery in the same way as the best current estimate of the value of the additional funds that would be paid for that resource.)

Then, even though the timing of the availability of user resources is at the mercy of the details of individual designs – and the laws of physics, chemistry, and astronomy – streams of user amounts of a resource are "indistinguishable" in the context of a design optimization problem if they have the same total (that is, integrated over the relevant time span) discounted present value.

The amounts of resources produced are also subject to variations with time due to the interaction of those same laws with maintenance policies. This variation is modeled as the product of a *nameplate size* that characterizes the subsystem and a production profile that contains a model of the variation with time. For example, the nameplate size for the electrical power system, for which the prime mover is the Sun, is most naturally expressed in terms of the solar energy collector area or in terms of the peak power under specified conditions. The production profile must account for degradation in performance due to yellowing of the plastic encapsulant that protects the photovoltaic cells or due to micrometeorite strikes on the solardynamic mirror surfaces, abrupt increases in performance that would result from block replacements of components or planned changes in capacity, and cyclic variations due to orbital decay between reboostings and the effects of the 11-year sunspot cycle.

The assertion that the amount of each resource supplied to users should be at least equal to a specified annual amount may thus be written in terms of present values.

Design alternatives to be compared can be characterized in terms of the amounts of resources produced, system-level design parameters, and subsystem-level design parameters. Thus, the SSFP design optimization problem may be stated as in Figure 1.

Find $N_j \forall j$ and $S_m \forall m$ and $V_{jn} \forall j, n$

To minimize LCC

Subject to $pv\{U_{jt}\} \geq pv\{GU_j \text{ for } 0 \leq t < Life\} \quad \forall j$ (1)

Where

N_j = nameplate size of subsystem j

S_m = m th system-level design parameter

V_{jn} = n th subsystem-level design parameter for subsystem j

LCC = estimated life-cycle cost for Space Station Freedom

U_{jt} = amount of resource j available for users at time t

GU_j = specified nominal annual user amount of resource j

$Life$ = operational lifetime of the station

$pv\{stream\}$ = integrated present value of the *stream* indicated within the braces

The symbol \forall means "for all appropriate values of," and may be read as "for all."

Figure 1. The SSFP Design Optimization Problem

Design choices are also constrained by the state of the art. However, among the major benefits of engaging in undertakings as grand as the development and construction of a space station are the planned – and serendipitous – improvements in the state of the art. The design optimization process must accommodate and should encourage such advances.

The problem could have been stated in a slightly more general form. That is, rather than characterizing all of the sizes by nameplate values that do not change throughout the lifetime, we could seek the optimal growth trajectory. We have investigated this formulation in unpublished earlier work: although the analytical solution is not significantly different from that which will be presented here, the computational requirements are significantly more severe. The biggest objection, however, is the difficulty of obtaining credible specifications of user requirements over the entire lifetime of the station. This topic is discussed in more detail in Section 7.

While the designs considered can be called "flat" because the nameplate sizes do not change over the lifetime, it is not necessary to assume that the amounts of resources produced are constant. The nameplate sizes can be multiplied by production profiles that take into account degradation, repair and replacement, changes in environmental conditions, and any other modelable causes of time variation in production. Thus,

$$X_{jt} = N_j f_{jt} \quad \forall j, \quad 0 \leq t < Life \quad (2)$$

where

$$\begin{aligned} X_{jt} &= \text{amount of resource } j \text{ produced in year } t \\ f_{jt} &= \text{production profile for resource } j \end{aligned}$$

The models of production profiles can be expected to be dependent on the design parameters, S_m and V_{jn} .

The amounts of resources available to users are simply what's left after consumption of resources by the station itself. That is,

$$U_{jt} = X_{jt} - Y_{jt} \quad \forall j, \quad 0 \leq t < Life \quad (3)$$

where

$$Y_{jt} = \text{amount of resource } j \text{ consumed by the station itself during year } t$$

The model for Y_{jt} can be expected to depend on the nameplate sizes of all subsystems, the amounts of each resource produced in year t , and, possibly, explicitly on t itself.

SECTION 2

SOLUTION OF THE MATHEMATICAL PROBLEM

When the "fixed size" perspective is taken for all resources (i.e., resource-producing subsystems) simultaneously, comparison of designs is very difficult (as discussed in Section 1), but solution of the mathematical problem is trivial; the complexity arises when the "fixed user amount" perspective is taken. Fortunately, it is not necessary to take either perspective for all resources simultaneously. We may – and will – ignore all of those resources for which the sizes are specified without losing any generality in treatment: in an implementation algorithm, it is merely necessary to set the appropriate nameplate sizes to their specified values. When nameplate sizes are determined in this way – by specification – they behave the same as if they were fixed design parameters.

In the derivation of the problem statement in Section 1, constraints were divided into two classes. Some were described as "screening conditions" that must be satisfied by any candidate design alternatives. Judgments and negotiations involving "higher authorities" were invoked for the application of these screens. (These issues include astronaut safety requirements, schedules, budgets, and the like.)

The remaining constraints specify nominal annual user amounts of resources. They contain descriptions of resource consumptions by possible subsystem designs and are the cornerstone of the analysis. We will use them to find balanced sets of nameplate sizes, which can then be used to compare technological alternatives.

If we insert Equations (2) and (3) into the constraint part of the problem statement, Equation (1), the specifications ($GU_j \forall j$) are connected with the nameplate sizes ($N_j \forall j$) and the consumption models ($Y_{jt} \forall j, t$):

$$pv\{N_j f_{jt} - Y_{jt}\} \geq pv\{GU_j \text{ for } 0 \leq t < Life\} \quad \forall j \quad (4)$$

The linearity of the present value operation allows restatement of Equation (4) as

$$N_j pv\{f_{jt}\} \geq \frac{GU_j}{crf} + pv\{Y_{jt}\} \quad \forall j \quad (5)$$

where the capital recovery factor, $crf = 1/pv\{1 \text{ for } 0 \leq t < Life\}$.

The present values of the production profiles are positive, and may be divided out of both sides without changing the sense of the inequality, giving

$$N_j \geq \frac{GU_j}{crf \cdot pv\{f_{jt}\}} + \frac{pv\{Y_{jt}\}}{pv\{f_{jt}\}} \quad \forall j \quad (6)$$

Before proceeding, let us digress to discuss the concept of a *design space*. Consider Figure 2, which shows a design space for the Space Station Freedom logo. The abscissa measures the relative size of one of the features of the logo, the stylized solar panels. The ordinate measures the relative size of another feature, the stylized habitat and laboratory modules. (These sizes are measured relative to a third feature, the stylized Earth.) Every point on this graph represents a possible design for the logo; some of the designs are "better" than others. Freedom's actual logo is shown in the middle column in the next to the bottom row.

Equation (6) can be interpreted graphically in terms of a space that contains all possible designs for Space Station Freedom. Instead of two dimensions, as we have for the logo in Figure 2, we need a dimension for each resource. Fortunately, we do not have to draw pictures that show all of the dimensions; two-dimensional slices will suffice for illustrations.

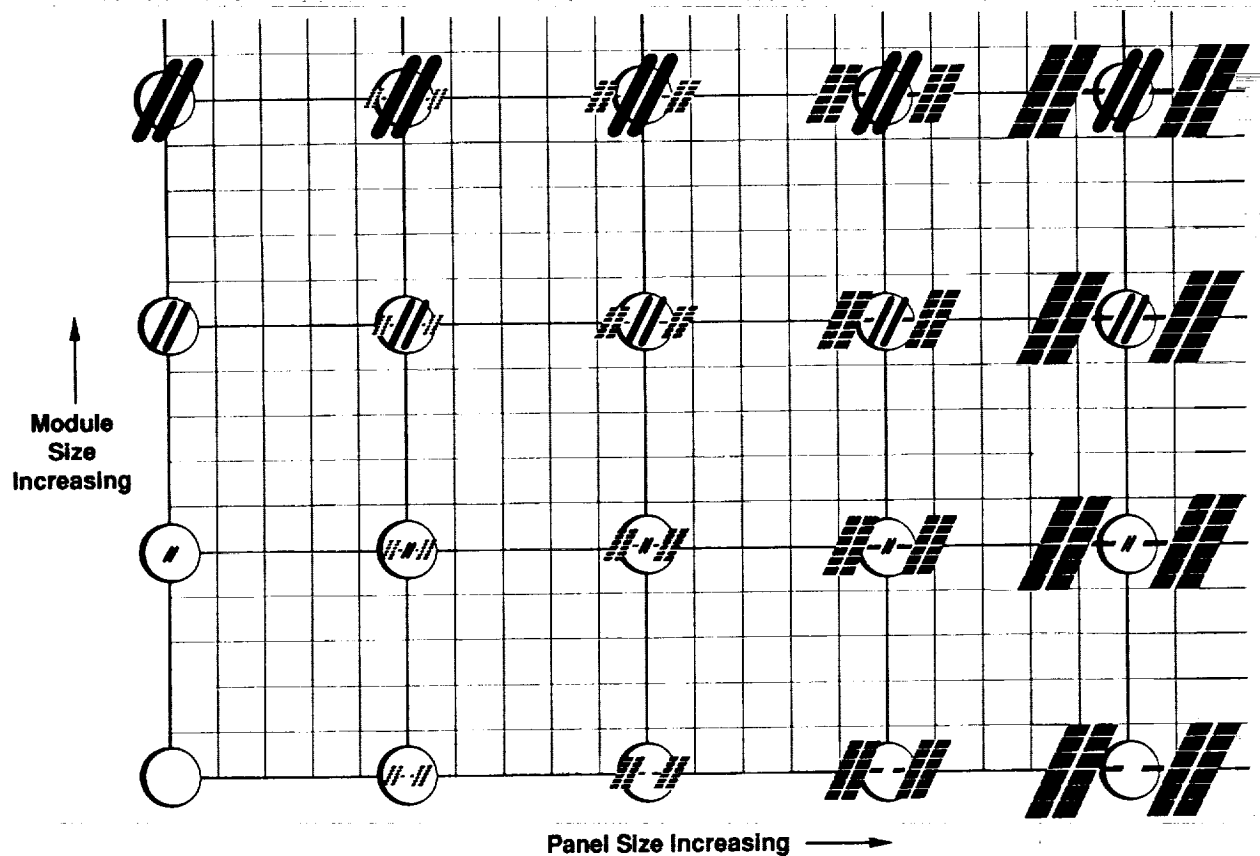


Figure 2. Design Space for the Space Station Freedom Logo

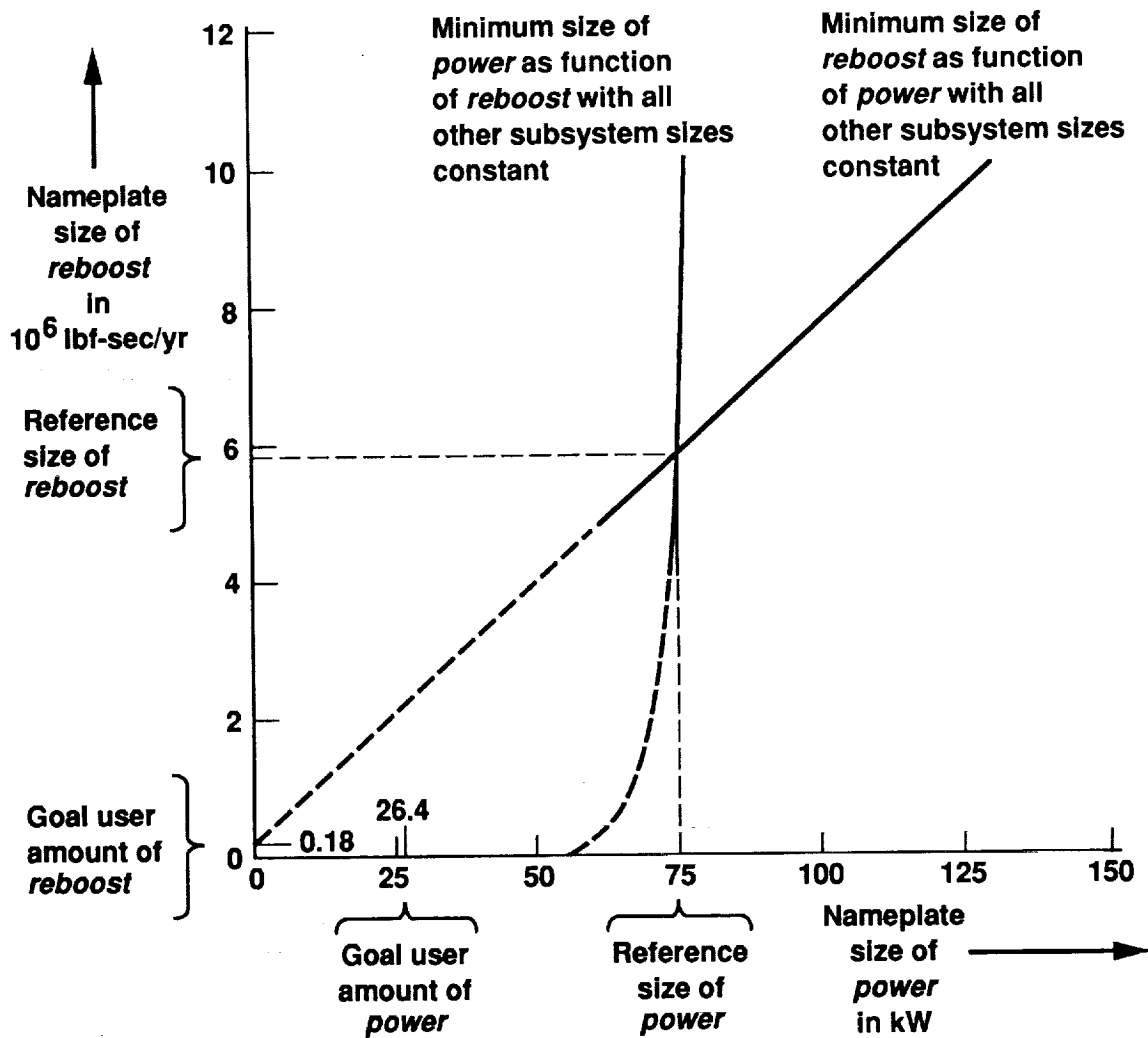


Figure 3. The *power, reboost* Slice of the SSF Design Space

Consider Figure 3, which shows a two-dimensional slice through the space of all possible designs for Space Station Freedom. The axes represent the nameplate sizes of two typical subsystems: on-board electrical *power*, and propulsion for *reboost* of the station to counter the effects of drag. Locate the specified user amount of *reboost* on the Y-axis: 0.18×10^6 lbf-sec/yr. Now, consider the consumption of *reboost* in the production of *power*. Freedom's source of electrical power is solar energy, which is rather diffuse. Large solar panels are required to collect enough energy to be useful. Although the Earth's atmosphere is extremely thin at orbital altitudes, it does produce some drag. The amount of drag depends strongly upon the size of the solar panels. (Drag also depends on time-varying factors, such as station altitude, the orientation of the panels, and the changing atmospheric density; so careful, detailed modeling is required to obtain good estimates of the average effect of size on *reboost* requirements.) Now, on Figure 3, keeping the nameplate sizes of all other subsystems constant, plot the nameplate size that the propulsion subsystem would have to have to satisfy the constraint given in Equation (6), where j is *reboost* and the *power* component of the consumption of *reboost* is shown on the abscissa. This line is labeled "Minimum size of *reboost*..."; part of it is dashed to suggest that there may be a minimum realizable size for one or both of the subsystems.

In one design of the *reboost* propulsion system, the propellants are produced by purifying, then electrolyzing, waste water: the resultant hydrogen and oxygen are then burned in the rockets to provide *reboost* thrust. The *power* requirements depend upon the amount of water that has to be electrolyzed, and hence on the size of the *reboost* subsystem. This relationship is also shown in Figure 3, with the roles of the abscissa and ordinate reversed. The resultant line is labeled "Minimum size of *power*..."

The relevant parts of the same two curves are shown in Figure 4. Only those designs in the indicated region between the two curves satisfy the constraint, Equation (6). Assuming that an increase in the size of a subsystem never decreases its consumption, the constraint lines will always have non-negative slopes. (They do not, however, have to be straight lines.)

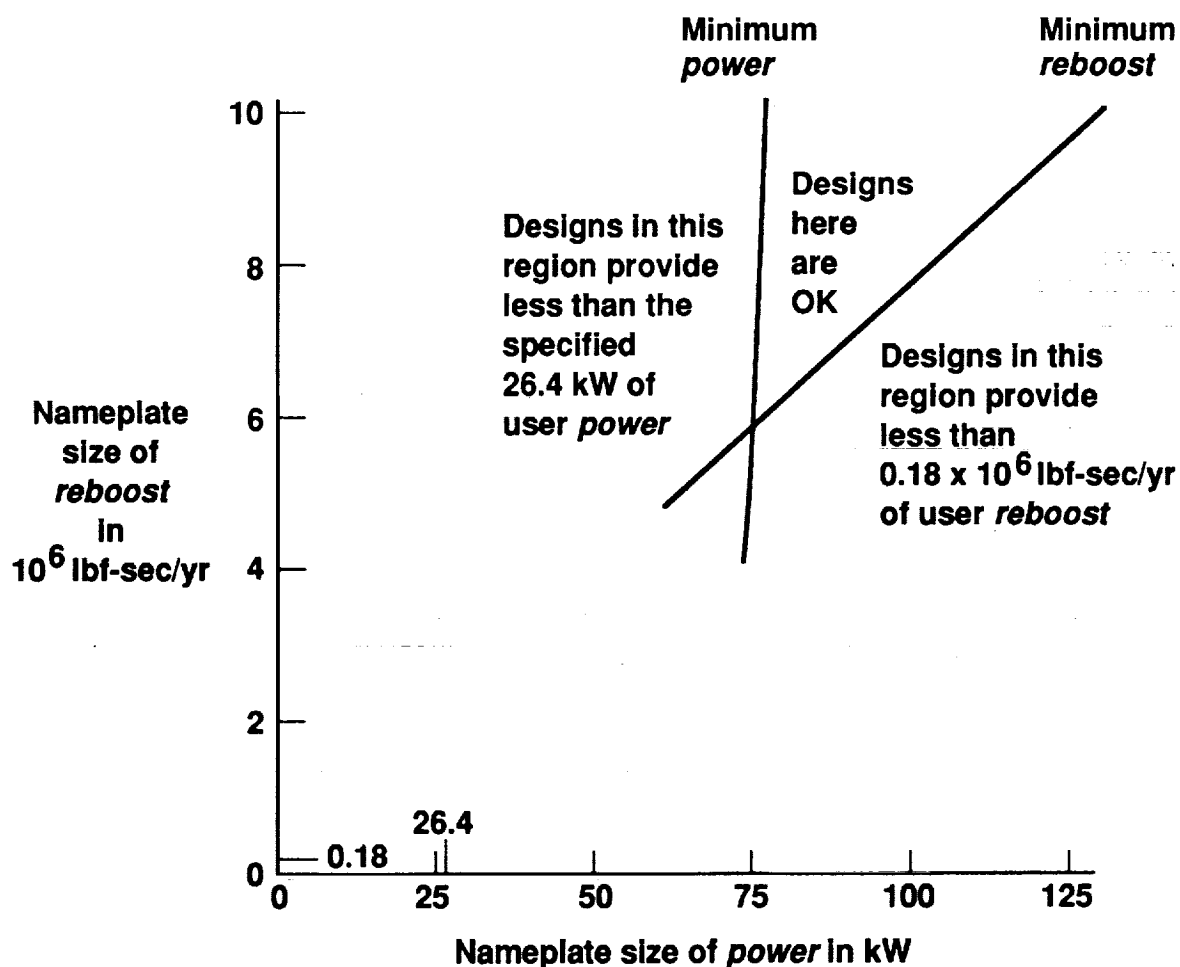


Figure 4. How Big is Big Enough?

The life-cycle cost is determined by the nameplate sizes of the subsystems. Figure 5 shows lines of constant life-cycle cost. If it is assumed that *LCC* never decreases when the nameplate size of a subsystem increases (with the nameplate sizes of all other subsystems held constant), the lines of constant *LCC* have a nonpositive slope and the constant *LCC*s associated with those lines decrease toward the origin. (That is, if one subsystem is bigger, the other must be smaller to keep the total cost the same.) Again, it is not necessary that these lines be straight.

Hence, the optimal set of sizes must be the one at the lower left corner of the region labeled "Acceptable Designs," as indicated in the figure. Moreover, it is apparent that the user specifications are all binding constraints. It can be shown, as is discussed in Section 7, that the precise requirement for the indicated point to be optimal is that the shadow prices associated with the constraints be non-negative. That is, it is not strictly necessary that consumptions and costs increase as nameplate sizes increase; some could decrease, but not by too much.

Is it coincidental that the optimal set of sizes corresponds to the reference design in this example (compare Figures 3 and 5)? Of course not; this is an inevitable consequence of alternating between the fixed-size and fixed-user-amount perspectives. At this point in the iterative design optimization process, the user-amount specifications have evidently just been set to the residual user amounts derived from the reference design.

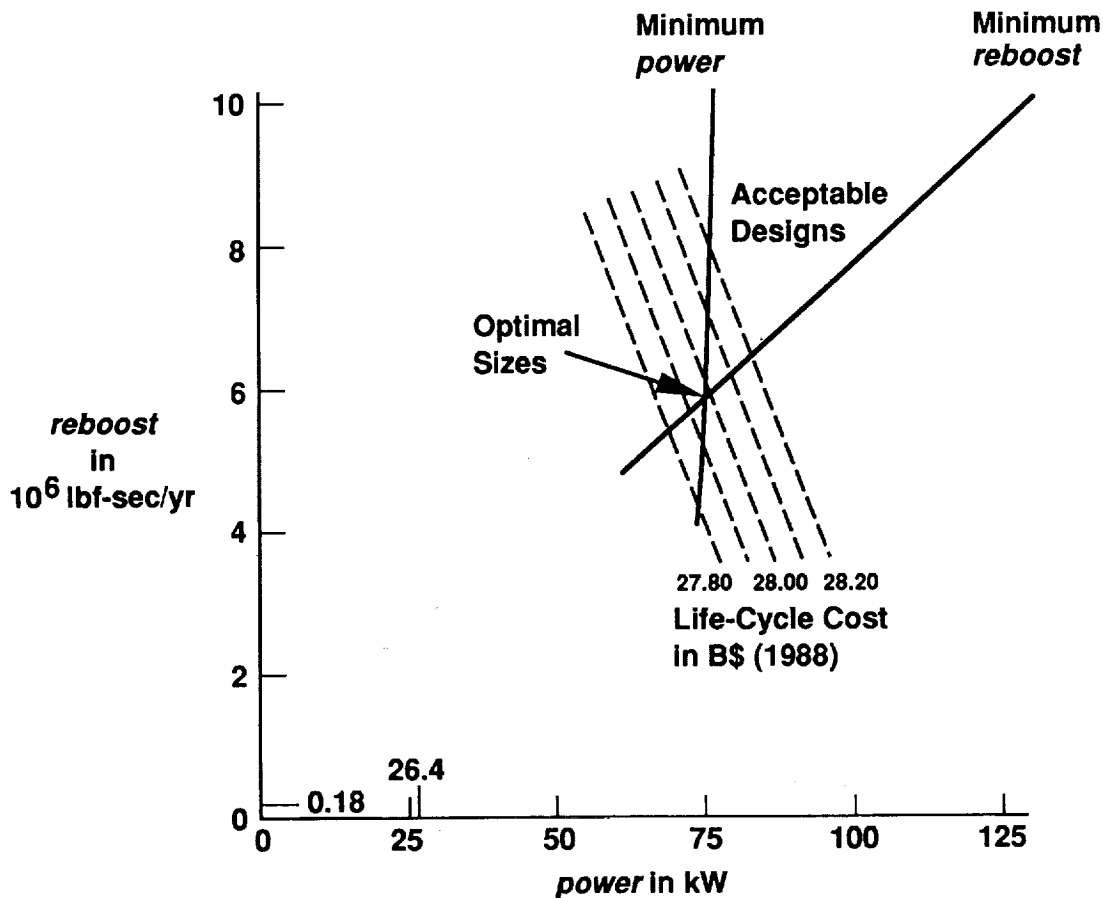


Figure 5. Solution of the Nonlinear Program

Suppose that an improved electrolysis unit, requiring 10 kW less power, is considered for the *reboost* subsystem. (This number was chosen to achieve visual separation of the lines on Figure 6, not because such a reduction in *power* requirements corresponds to any currently proposed design change.) As shown in Figure 6, the "Minimum *power*" curve shifts left by 10 kW, giving a new set of optimally balanced sizes. The new size of the *power* subsystem is more than 10 kW smaller than the old one, because the smaller *power* subsystem requires less reboost, which requires still less power, and so on. This series of "design ripples" converges very rapidly.

When housekeeping power consumption decreased by 10 kW, the *LCC* decreased by 260 M\$(1988). The associated *marginal cost* of 26 M\$(1988)/kW expresses the way that additional (or reduced) use of the resources affects the station's life-cycle cost, and can be used directly in design trades. For validity, the cost and consumption functions must be close enough to linear that marginal costs do not vary much in the parts of the design space being investigated.

We may also note that a lump sum change in housekeeping consumption, as in the 10-kW illustration just given, has exactly the same effect as a similar change in user requirements. Most design changes that lead to changes in housekeeping consumption will, however, affect the slope of the consumption curve in addition to its intercept.

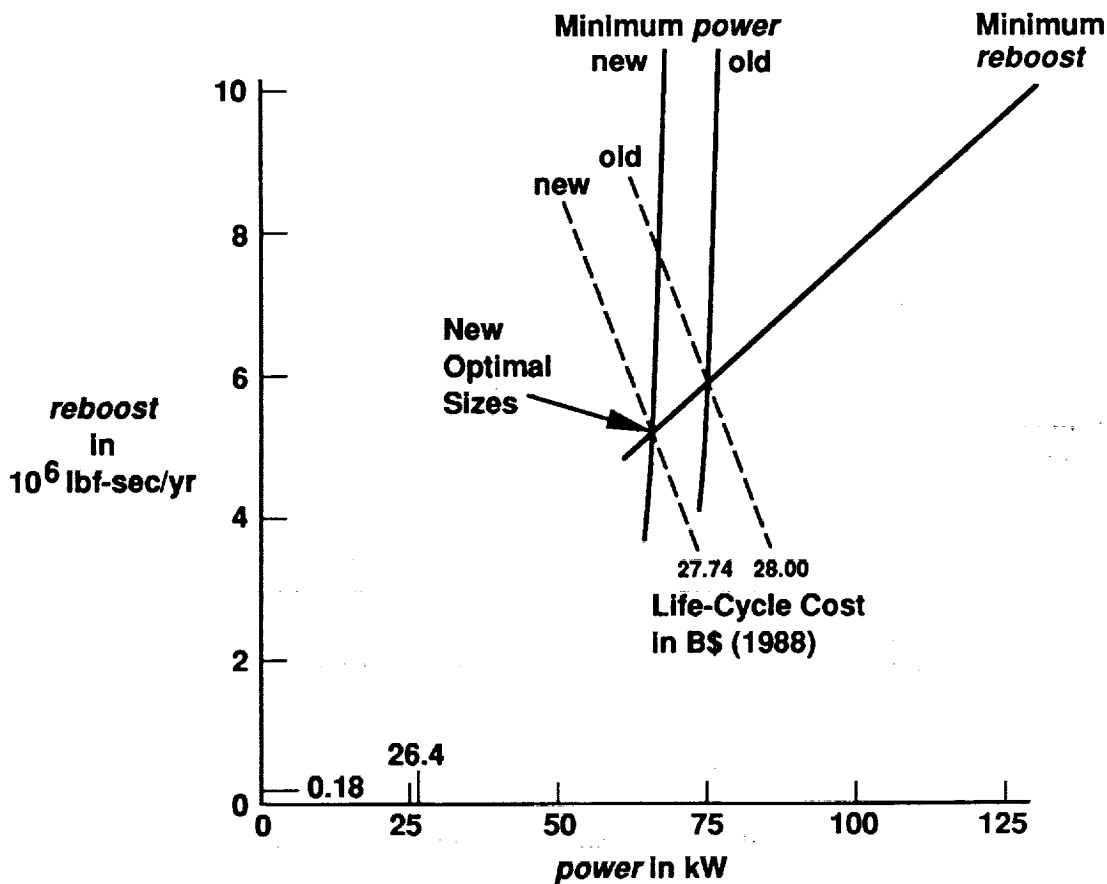


Figure 6. New Sizes for New Technology

SECTION 3

DECENTRALIZATION OF THE PROCESS

As demonstrated in Section 2, the design optimization problem stated in Section 1 could be solved by a centralized authority – if it had all of the necessary information. In reality, however, the necessary information is widely dispersed, both geographically and organizationally. The centralized authority of Space Station Freedom does not and cannot have all of the necessary information, some of which is proprietary or, perhaps, even yet to be discovered by the organizations responsible for designs of the subsystems. It would be better if the problem could be decentralized, so that system-level decisions are made by the central authority, but subsystem-level decisions are made at a lower level, as independently of the decisions made for other subsystems as is practical. The purpose of this section is to develop that decentralized statement of the problem.

The decentralized process consists of alternately solving a system-level problem and a set of subsystem-level problems. The system-level problem relies on the presumption that the state of the art for each subsystem is enveloped by the consumption and cost curves of the type suggested in Figure 5. The problem is then to find the optimally balanced set of nameplate sizes and the optimal values of system-level design parameters.

The subsystem-level problem relies on the presumption that the effect of housekeeping consumption of resources on life-cycle cost is completely captured by the marginal costs which the central authority computes for those resources. The problem then is to select a design – at the size determined by the solution to the system-level problem – that minimizes the subsystem's contribution to life-cycle cost. That contribution has two parts: the obvious, explicit part, and an implicit part composed of "purchases" and "sales" of resources at their marginal costs. The results are reported back to the system level as new consumption, cost, and production formulas, presented as functions of subsystem nameplate sizes. Those formulas represent the envelope of best designs for nameplate sizes near the size specified by the central authority.

The system-level and subsystem-level problems are designed so that the optimization objectives are aligned. Then, improvements made in one area will not be undone by improvements in another. If the production, cost, and consumption formulas are prepared in sufficient detail and with sufficient accuracy, and are not strongly nonlinear with nameplate sizes, marginal costs will show but little variation throughout the relevant part of the design space, and convergence of the process can be expected to be very rapid.

Although the process just described encompasses only system-level and subsystem-level design decisions, it can actually be applied quite easily at more detailed levels as well: designers simply need to treat on-board resources as if they had to pay for them at their marginal costs. Decisions made on this basis will deviate from optimality only to the extent that marginal costs vary in different parts of the relevant portion of the design space. Stable marginal costs can be expected unless there are design breakthroughs, strong nonlinearities in the design functions, or significant changes in the station design concept.

1. The first part of the document is a letter from the President of the United States to the Congress, dated January 3, 1862. It is a very important document, as it contains the President's annual message to Congress, which is a key part of the executive branch's communication with the legislative branch.

2. The second part of the document is a report from the Secretary of the Interior, dated January 10, 1862. It contains information about the state of the Department of the Interior, including the status of the various bureaus and the progress of the work of the department.

3. The third part of the document is a report from the Secretary of the Treasury, dated January 15, 1862. It contains information about the state of the Department of the Treasury, including the status of the various bureaus and the progress of the work of the department.

4. The fourth part of the document is a report from the Secretary of the War, dated January 20, 1862. It contains information about the state of the Department of the War, including the status of the various bureaus and the progress of the work of the department.

5. The fifth part of the document is a report from the Secretary of the Navy, dated January 25, 1862. It contains information about the state of the Department of the Navy, including the status of the various bureaus and the progress of the work of the department.

6. The sixth part of the document is a report from the Secretary of the Army, dated February 1, 1862. It contains information about the state of the Department of the Army, including the status of the various bureaus and the progress of the work of the department.

7. The seventh part of the document is a report from the Secretary of the Marine Corps, dated February 5, 1862. It contains information about the state of the Department of the Marine Corps, including the status of the various bureaus and the progress of the work of the department.

8. The eighth part of the document is a report from the Secretary of the Coast and Geodetic Survey, dated February 10, 1862. It contains information about the state of the Department of the Coast and Geodetic Survey, including the status of the various bureaus and the progress of the work of the department.

9. The ninth part of the document is a report from the Secretary of the Smithsonian Institution, dated February 15, 1862. It contains information about the state of the Department of the Smithsonian Institution, including the status of the various bureaus and the progress of the work of the department.

10. The tenth part of the document is a report from the Secretary of the United States Geological Survey, dated February 20, 1862. It contains information about the state of the Department of the United States Geological Survey, including the status of the various bureaus and the progress of the work of the department.

SECTION 4

DECOMPOSITION

We will use the technique known as "Lagrangian relaxation" to separate the overall design optimization problem into system-level and lower-level parts. We will construct a modified objective function, called the Lagrangian, by adding weighted penalties to the life-cycle cost for each constraint that might be violated. The solution then satisfies a set of first-order conditions: extrema occur where the first partial derivatives of the Lagrangian, simultaneously taken with respect to each of the decision variables, vanish. The weights, known as Lagrange multipliers, are treated as decision variables so that the first-order conditions include the original constraints. When the constraints are satisfied, they incur no penalties, so the Lagrangian is equal to the original objective function.

Refer to the problem statement, Figure 1. Using the observation that all of the constraints are binding, the Lagrangian is

$$L = LCC + \sum_j \lambda_j (pv\{GU_j\} - pv\{U_j\}) \quad (7)$$

where the symbols λ_j represent the to-be-determined Lagrange multipliers. As in Section 2, use Equations (2) and (3) to connect the input data with the nameplate sizes and with the consumption models in this equation, obtaining

$$L = LCC + \sum_j \lambda_j (pv\{GU_j\} - pv\{N_j f_{jt} - Y_{jt}\}) \quad (8)$$

The first-order conditions are that the partial derivatives of L with respect to the Lagrange multipliers ($\lambda_j \forall j$), the nameplate sizes ($N_j \forall j$), the system-level design parameters ($S_m \forall m$), and the subsystem-level design parameters ($V_{jn} \forall j,n$) all vanish at the solution. We will deal with each of the resulting sets of conditions in turn, and identify the implications for decentralization.

4.1 RESOURCE SIZES

Zeroing the derivatives with respect to the Lagrange multipliers reproduces the constraint equations:

$$\frac{\partial L}{\partial \lambda_j} = pv\{GU_j\} - pv\{N_j f_{jt} - Y_{jt}\} = 0 \quad \forall j \quad (9)$$

These equations may be restated with the nameplate sizes explicitly on the left-hand side (and implicitly within the right-hand side) as follows:

$$N_j = \frac{GU_j}{crf \cdot pv\{f_{jt}\}} + \frac{pv\{Y_{jt}\}}{pv\{f_{jt}\}} \quad \forall j \quad (10)$$

(Note the similarity to Equation (6).)

Solution of this set of equations (for $N_j \forall j$) is a job for the central authority, for it requires knowledge of all of the user specifications ($GU_j \forall j$), the production profiles ($f_{jt} \forall j, t$), and the consumption functions ($Y_{jt} \forall j, t$).

4.2 LAGRANGE MULTIPLIERS (IMPLICIT PRICES)

Zeroing the derivatives of the Lagrangian with respect to the nameplate sizes produces equations that can be solved for the Lagrange multipliers:

$$\frac{\partial L}{\partial N_i} = \frac{\partial LCC}{\partial N_i} - \sum_j \lambda_j \left(pv \left\{ \delta_{ji} f_{jt} - \frac{\partial Y_{jt}}{\partial N_i} \right\} \right) = 0 \quad \forall i \quad (11)$$

where δ_{ji} is the Kronecker delta, equal to unity if j and i are the same, zero otherwise. With some rearrangement of terms, Equation (11) may be restated as:

$$\lambda_i = \frac{1}{pv\{f_{it}\}} \frac{\partial LCC}{\partial N_i} + \sum_j \lambda_j \frac{pv \left\{ \frac{\partial Y_{jt}}{\partial N_i} \right\}}{pv\{f_{it}\}} \quad \forall i \quad (12)$$

The central authority must also solve these equations (for $\lambda_i \forall i$), as they depend upon system-level information.

4.3 SYSTEM-LEVEL DESIGN PARAMETERS

Zeroing the derivatives of the Lagrangian with respect to the system-level design parameters produces equations that can, in principle, be solved simultaneously with Equations (9) and (11) for the optimal values of those parameters:

$$\frac{\partial L}{\partial S_m} = \frac{\partial LCC}{\partial S_m} + \sum_j \lambda_j \quad pv \left\{ \frac{\partial Y_{jt}}{\partial S_m} \right\} = 0 \quad \forall m \quad (13)$$

These equations must also be solved (for $S_m \forall m$) by the central authority.

It should be noted that different values of system-level design parameters (such as station altitude at shuttle rendezvous) may lead to qualitatively different system designs, which warrant consideration at a level of management higher than the central authority, as discussed in Section 1. Thus, the central authority must often find the appropriate values of the system-level design parameters by a process other than that of solving Equation (13).

4.4 SUBSYSTEM-LEVEL DESIGN PARAMETERS

Zeroing the derivatives of the Lagrangian with respect to the subsystem-level design parameters for subsystem i produces the set of equations to be solved by the designers of subsystem i :

$$\frac{\partial L}{\partial V_{in}} = \frac{\partial LCC}{\partial V_{in}} + \sum_j \lambda_j \left(-N_j \text{pv} \left\{ \frac{\partial f_{jt}}{\partial V_{in}} \right\} + \text{pv} \left\{ \frac{\partial Y_{jt}}{\partial V_{in}} \right\} \right) = 0 \quad \forall n, \forall i \quad (14)$$

If these equations do not depend upon information that is specific to the designs of other subsystems, then they can be solved by the producer of resource i more or less in isolation. Mathematically, these *separability conditions* are that the partial derivatives of the expression between the equals signs in Equation (14) with respect to each of the variables that describe the other subsystems (that is, N_l and V_{lq} for all $l \neq i$ and for all values of q appropriate to subsystem l) must be identically zero for all of the designs in a neighborhood of the optimal design in the design space. Specifically, the derivatives are as follows:

$$\begin{aligned} \frac{\partial}{\partial N_l} &\Rightarrow \frac{\partial^2 LCC}{\partial N_l \partial V_{in}} + \sum_j \lambda_j \left(-\delta_{jl} \text{pv} \left\{ \frac{\partial f_{jt}}{\partial V_{in}} \right\} + \text{pv} \left\{ \frac{\partial^2 Y_{jt}}{\partial N_l \partial V_{in}} \right\} \right) \equiv 0 \quad \forall l \neq i, n \\ \frac{\partial}{\partial V_{lq}} &\Rightarrow \frac{\partial^2 LCC}{\partial V_{lq} \partial V_{in}} + \sum_j \lambda_j \left(-N_j \text{pv} \left\{ \frac{\partial^2 f_{jt}}{\partial V_{lq} \partial V_{in}} \right\} + \text{pv} \left\{ \frac{\partial^2 Y_{jt}}{\partial V_{lq} \partial V_{in}} \right\} \right) \equiv 0 \quad \forall l \neq i, n, q \end{aligned}$$

The fact that these conditions must hold throughout a neighborhood in the design space implies that each term in the above equations must be equal to zero:

$$\begin{aligned} \frac{\partial^2 LCC}{\partial N_l \partial V_{in}} &= 0 \quad \forall l \neq i, \forall n \\ \frac{\partial f_{lt}}{\partial V_{in}} &= 0 \quad \forall t, \forall l \neq i, \forall n \\ \frac{\partial^2 Y_{jt}}{\partial N_l \partial V_{in}} &= 0 \quad \forall j, t, \forall l \neq i, \forall n \\ \frac{\partial^2 LCC}{\partial V_{lq} \partial V_{in}} &= 0 \quad \forall l \neq i, \forall q, \forall n \\ \frac{\partial^2 f_{jt}}{\partial V_{lq} \partial V_{in}} &= 0 \quad \forall j, t, \forall l \neq i, \forall q, \forall n \\ \frac{\partial^2 Y_{jt}}{\partial V_{lq} \partial V_{in}} &= 0 \quad \forall j, t, \forall l \neq i, \forall q, \forall n \end{aligned} \quad (15)$$

If any of these separability conditions are not satisfied for a design parameter, that parameter must be categorized as a system-level design parameter, and should be dealt with by the central authority. Hence, the separability conditions are in principle always satisfied, by definition. (In practice, it may be convenient to assign the choices of some parameters that do not satisfy Equations (15) to a subsystem anyway. If those equations are nearly satisfied, the very rapidly convergent decentralized design optimization process will merely converge a little less rapidly.)

4.5 DEFINITION OF SUBSYSTEM LIFE-CYCLE COST

Equations (14) are conditions that are satisfied in the optimal design, and do not require information unavailable to the designers of subsystem i . The decentralized subsystem design optimization problem in Figure 7 is defined so that it satisfies exactly the same set of conditions.

Given $N_i, \lambda_j \forall j, S_m \forall m,$

Find $V_{in} \forall n$

To minimize $LCC_i = ExplCost_i + ImplCost_i - ImplRevenue_i$

Subject to the screening conditions discussed in Section 1

Where

$LCC_i \triangleq$ "life-cycle cost for subsystem i "

$ExplCost_i$ = explicit cost; subsystem i 's contribution to the station's life-cycle cost at the optimal design in the design space, with the designs (and sizes) of all other subsystems held constant
 $= LCC(N_j \forall j) - LCC(N_j \forall j \neq i, N_i = 0)$

$ImplCost_i$ = implicit cost; the effect of subsystem i on the station's life-cycle cost due to its net consumption of resources
 $= \sum_j \lambda_j pv\{A_{jit}\}$

A_{jit} = amount of resource j consumed by subsystem i during year t

$ImplRevenue_i$ = implicit revenue; the effect of subsystem i on the station's life-cycle cost due to its production of resource i
 $= \lambda_i N_i pv\{f_{it}\}$

Figure 7. The Design Optimization Problem for Subsystem i

SECTION 5

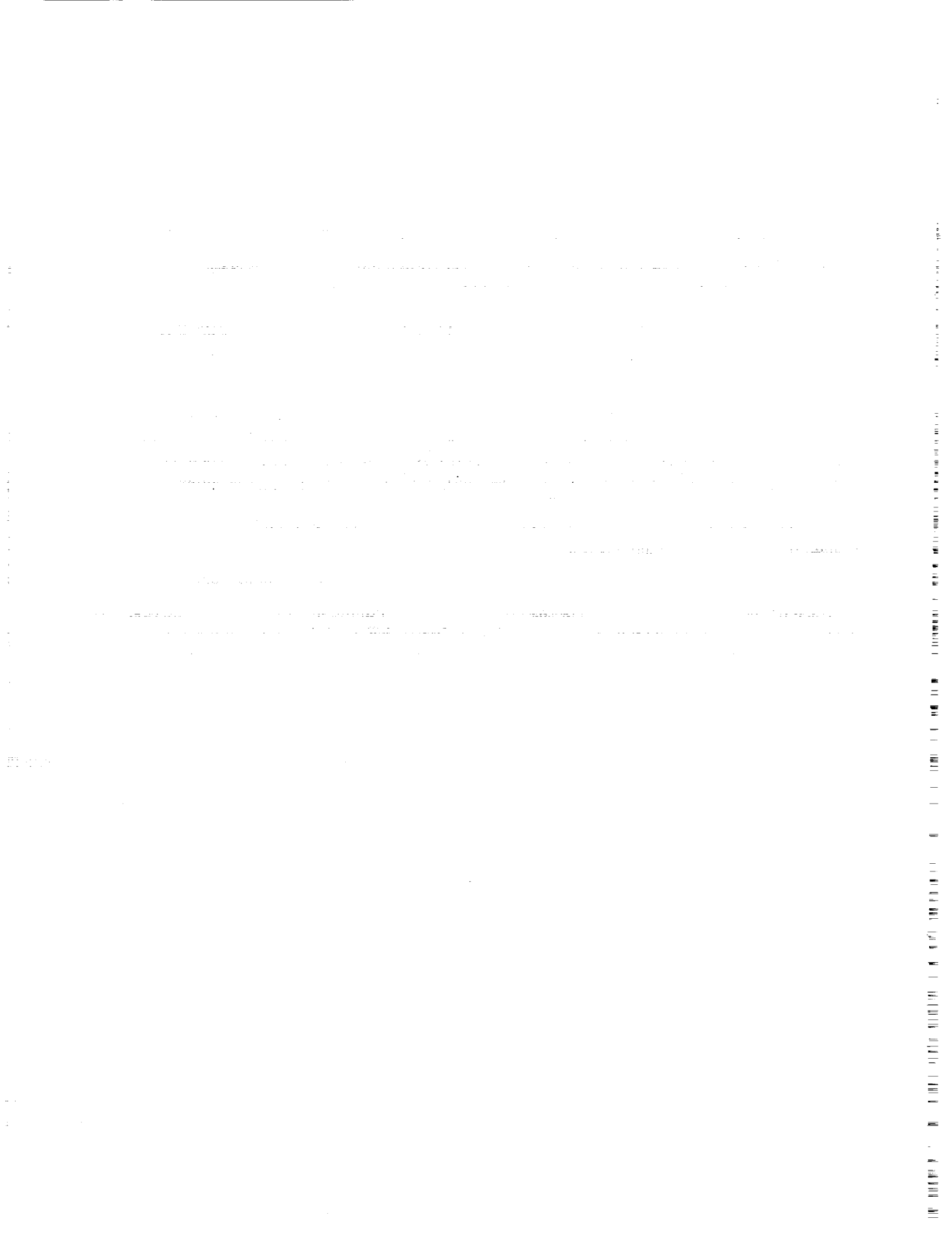
THE SUBSYSTEM-LEVEL DESIGN OPTIMIZATION PROBLEM

As stated in Figure 7 in the previous section, the subsystem designers' problem can be described as finding designs that respond optimally to the information provided by the central authority. In particular, subsystem designers must treat the *implicit* costs and revenues associated with their designs as seriously as they do the *explicit* costs.

Program management should establish criteria for design reviews that will ensure that this will happen. That is, subsystem managers' budgetary performance should be judged on subsystem life-cycle cost as defined in Figure 7.

Subsystem designs should be based on the nameplate sizes specified by the central authority and must satisfy all of the screening conditions. In addition, formulas that describe the envelope of optimal designs associated with the supplied set of implicit prices are needed for the continued application of the overall process. The "design model" part of the SDTM computer program, described in Section 6, can facilitate preparation of these formulas from basic design databases containing mass estimates, power requirements, mean times between failures, and similar information about equipment items.

If the feasible designs for a subsystem are well understood and the implicit prices provided by the central authority do not change, the cost and consumption formulas that describe the subsystem's state of the art to the central authority do not change. When none of the cost or consumption formulas change, none of the implicit prices change, and the process converges.



SECTION 6

THE SYSTEM-LEVEL DESIGN OPTIMIZATION PROBLEM

The system-level problem is to use the description of the design space which is provided by the cost and consumption formulas to find

- (1) the optimal balance of nameplate sizes, by solving Equation (10),
- (2) the implicit prices by solving Equation (12), and
- (3) the optimal values of system-level design parameters, by solving Equation (13).

These results are communicated to the subsystem designers for the next step in the iterative decentralized SSFP design optimization process, as discussed in the previous section.

A computer program called SDTM (System Design Tradeoff Model) has been developed to help with the system-level problem. SDTM contains two parts. The "design model" part uses detailed models of the performance, consumption, and logistics associated with subsystem designs to construct many of the cost and consumption formulas from data describing equipment items, assembly flights, astronauts, and so on. (Thus, SDTM may be useful in solving subsystem-level problems.) The "core" part solves Equations (10) and (12) for given values of system-level design parameters.

The remaining subsections in this section cover some of the details associated with the core part of SDTM. A more complete description is given in Appendix A.

6.1 SIZING AND (IMPLICIT) PRICING

Solution of Equations (10) for the nameplate sizes is algorithmically very easy. Consider Figure 6. The task is that of moving from the reference design – the point at which the "Minimum *reboost*" line intersects the "Minimum *power* old" line – to the "Minimum *power* new" intersection, though in more dimensions. A Gauss-Seidel procedure has been found to be very effective: initialize the N_j to the nameplate sizes associated with the reference design. Use those values in the right-hand sides of Equations (10) to compute updated estimates of the N_j . Use the updated values of the N_j as soon as they are available. Continue doing so until convergence is achieved. For technically feasible designs, convergence is rapid; infeasible technology (or typographical errors during data input, perhaps) could cause the algorithm to diverge or to fail to converge.

Equations (12) can also be easily solved by a Gauss-Seidel procedure. The first term on the right-hand side provides satisfactory starting values for the λ_i .

6.2 COSTING

The effects of the nameplate sizes of the subsystems on the station's life-cycle cost are an essential part of the subsystem-level optimization problem, as shown in Figure 7.

Cost estimation (and accounting) within the Space Station Freedom Program is based on a tree-like work breakdown structure. The top level or "root" of this tree represents NASA Headquarters in Washington, D.C., and is the "highest level of SSFP management," as discussed in Section 1. The next level is located in Reston, Virginia, and corresponds to the "central authority" discussed in Section 3 and afterward. Most of the subsystem-level design decisions are made by NASA field centers, at the next level of the tree. The field centers have prime contractors, the primes have subs, and so on.

Life-cycle cost is obtained in SDTM by accumulating cost estimates for leaves and branch points in the work breakdown structure tree, summing across cost estimation categories. Cost estimates are spread over time, marked up by wrap fractions, and rolled up the tree. The life-cycle cost is obtained by taking the present value of the costs which have been rolled all the way to the root of the tree.

Estimates of the way that the life-cycle cost depends on the amounts of resources produced are obtained by comparing life-cycle cost estimates for station designs that produce different amounts of resources. (These estimates are needed for the first term on the right-hand side of Equation (12).)

SECTION 7

EXTENSIONS

We have addressed several additional issues. These analyses are presently documented only as internal JPL technical memoranda. The substance of these studies is briefly summarized below.

7.1 NON-NEGATIVITY OF IMPLICIT PRICES

The identification of the optimal point in the design space (see Figure 5) in Section 2 relied upon the assumption that consumptions and costs of subsystems never decrease as nameplate sizes increase. Mere economies of scale do not violate this assumption unless they are so extreme that they cause total consumptions or costs — not just average consumptions or costs — to decrease. These conditions are sufficient, but are tighter than is necessary for that design to be optimal. Some of the incremental consumptions and costs *can* be negative, if they are small enough when evaluated at the indicated design point.

How small is "small enough"? Generally, beneficial byproducts are produced in small enough quantities to satisfy the requisite conditions. Reaction mass which is ejected at high velocity by the *reboost* subsystem, for example, reduces the mass that must be returned to Earth, but it is not desirable to make the *reboost* system larger just to obtain that benefit.

The precise conditions for the identified point to correspond to the optimal sizes can be found by continuing the analysis of the Lagrangian in Section 4. The second partial derivatives produce the second-order conditions associated with the optimal solution. Skipping the math, we have: The marginal costs of all resources, evaluated at the optimal design, must be non-negative. (A negative marginal cost would mean that the life-cycle cost will *decrease* if a *larger* amount of the resource is used, either by the station itself or by the station's customers.)

7.2 AGGREGATION AND RESOLUTION OF RESOURCES

A station design has been described here in terms of some number of resources. A fundamental issue in preparing such a description is how finely the design should be resolved. Should *power*, for example, be represented as a single resource, or should *power generation*, *energy storage*, and *power management and distribution* be distinguished?

The best choice depends, of course, on what is to be done with the results of analysis. Congress, for example, might be interested in a monolithic description, with the station characterized by a single variable such as *crew size*. Station customers might be interested only in those resources that will be made available for their use. Designers of subsystems are interested in the designs of other parts of the station only in a general way, but would like to be able to represent their designs in considerable detail.

There are two issues to be addressed: how a finely resolved description should be aggregated, and how a highly aggregate description should be resolved. In either case, the results should be consistent. That is, whether a subsystem is aggregated or resolved,

- (1) it should place the same demands for all other resources, and
- (2) the use of the resource or resources that it provides should have the same effect on life-cycle cost.

7.2.1 Aggregation

In an aggregate description, several resources are represented by a combined resource. It must be assumed when analyzing the aggregate description that marginal increments to the component resources occur in known proportions. If that assumption is poor, then the nameplate sizes that are found by the sizing algorithm will not correspond to the minimum life-cycle cost. If, on the other hand, the assumption is close to the truth, the sizes found will indeed be close to the sizes that would have been found by analysis of the more fully resolved description.

The keys to aggregation are to determine those fixed proportions and the marginal costs of the resource-producing subsystems to be combined. The marginal cost of the resource produced by this collection of component subsystems is a sum of the marginal costs of the component resources, weighted by the fixed proportions. Consumptions by the composite subsystem are then just a weighted sum of the consumptions by the components. Consumptions of the composite resource are also a weighted sum, but the weighting must be by marginal costs instead of by the fixed proportions. Care should be taken to distinguish between average and marginal consumptions, as they can be significantly different.

7.2.2 Resolution

Suppose that, instead of proceeding from a finely resolved description to an aggregated one, it is desired to resolve a resource into its component parts.

A more finely resolved description requires considerably more data. The temptation to use the existing aggregate data to simultaneously reduce the requirements for new data and ensure consistency should, however, be resisted, in favor of the presumably greater accuracy that can be obtained for resolved data. The results should be reaggregated if an aggregate description is still desired.

7.3 DESIGN TRAJECTORIES AND OPTIMAL GROWTH

SDTM assumes that the design of each subsystem can be characterized by its nameplate size; and that the required amount of a resource can be characterized by either the specified nominal annual user amount or the specified nameplate size. Time-dependent variations in cost and productivity are described by input profiles, and the input housekeeping consumption

formulas can depend explicitly on time as well. Preplanned growth in capabilities can thus be readily incorporated into the station description.

But what is the optimal growth path? It is a straightforward task to extend the problem formulation to find an approximately optimal design trajectory. The fundamental objection to doing so, however, is that required amounts of user resources must be specified for every year of the station's lifetime. Most of the customers of the station are expected to be scientists conducting experiments. The particulars of those experiments, hence their preferred balance of resources, will depend on the results obtained from earlier experiments. Thus, while time-phased requirements could be stated, their validity would be seriously suspect. It is far better to have designs with inherent flexibility. Unless uncertainty is dealt with explicitly (see the next subsection), flexibility is a meta-issue like crew safety, and must be dealt with by a higher level of management.

The extension of the problem statement requires describing both the specifications and the optimized sizes as functions of time, rather than as representative single numbers. To be practical, time should be resolved to the resupply interval (rather than continuously). Because the costs of making something bigger to begin with are often quite different than the costs of adding on, two kinds of growth should be identified: changes in the initial sizes of subsystems, and changes during operations. The optimization problem should be solved for both sets of changes simultaneously. Very similar algorithms can be used, but the problem is significantly larger: if there are S subsystems and T time periods, there are $S \times T$ size changes to be found, in addition to S nameplate sizes.

The resultant optimal design trajectory is subject to an important caveat: the life-cycle cost objective function, while assumably convex with respect to nameplate sizes (or initial sizes), is not necessarily convex with respect to the later size changes. That is, due to possible economies of scale, it may be cheaper to combine indicated size increases, adding some capacity before it is needed. The analysis would indicate merely when the increases in capacity are needed. This caveat is significantly weakened, however, by the observation that suspected nonconvexities of this sort should be few and easy to recognize. When they do appear, they can be readily analyzed on a case-by-case basis.

7.4 UNCERTAINTIES

A considerable amount of data is required to describe a station design; very little of that data is known with high accuracy. In fact, even what we will *want* the station to be able to do is not really knowable with high accuracy in advance. Evaluations of design alternatives should take all of these uncertainties into account – but how?

Sensitivity analysis can provide some insight into the consequences of uncertainty, and is easy to perform with a tool like the SDTM computer program. The analyst simply assumes that all data, with the exception of one design parameter, are known with certainty, and then inspects the analytical results associated with variations in that parameter.

The process could be made more sophisticated, at the cost of obtaining a lot more data and using a lot more computer time, by replacing some or all of the input data values by probability distributions, and then using Monte Carlo runs to determine how the distributions of the analytical results vary with the parameter. It can, however, be quite difficult to know what to do with all of this information – for example, is it better or worse to have a cost estimate with a higher mean but smaller variance?

An intriguing alternative approach is to formulate the problem statement in terms of the parameters of the probability distributions. The following development sequence could be used:

Let C denote the levelized (see Appendix B) equivalent of the total cost of completing the project. That is, C is calculated so that the present value of a stream of payments of C base-year dollars each year until the end of the project's life equals the present value of all costs yet to be spent throughout the project's lifetime. (Sunk costs may be included in the computation of C if desired.) The levelized value, rather than the present value, is used to reduce the effect of uncertainty about the project lifetime.

The value of C depends upon the decision variables, which are the nameplate sizes of subsystems and the nominal values of system-level and subsystem-level design parameters. The actual sizes of subsystems and values of design parameters, as well as the validity of the cost-estimating relationships themselves, are subject to uncertainty. Consequently, C will be stochastic.

Let C° denote the predicted maximum (with risk α_C) real-levelized cost of completing the project. That is,

$$\Pr\{C > C^\circ\} < \alpha_C$$

where

α_C = (exogenously specified) cost risk

C° = predicted maximum (with risk α_C) real-levelized cost of completing the project

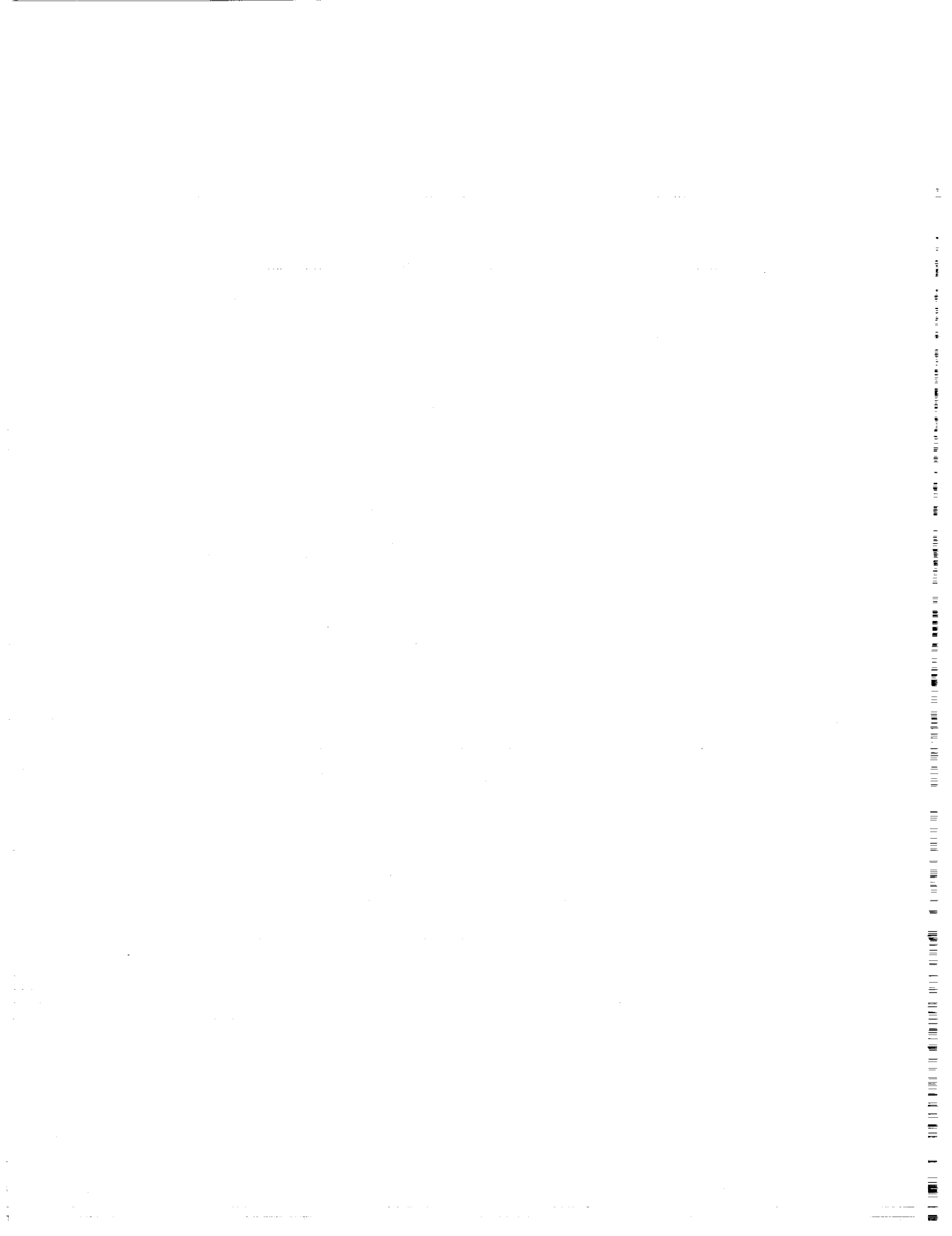
User amount specifications require two numbers for each resource: a nominal annual user amount, GU_j , as in Section 1 (or a year-by-year user amount as discussed in the previous subsection) and a user-amount risk level, α_j , which is the probability that the actual amount of resource j to be made available to users will be less than the specified amount in some year. This is an availability constraint. Calculations could be based on capacity expansion models (which use load-duration curves and other high-quality, hard-to-obtain data) or on weaker models of the interaction between stochastic supply and stochastic demand. If capacity expansion models are used, the α_j can be interpreted as the "acceptable loss of load probabilities."

The values of the desired nominal annual user amount specifications, the GU_j , are themselves uncertain. This uncertainty could be folded into the availability risk analysis just described. Doing so, however, would mix the supply-side risk analysis, for which high-quality data can

conceivably be obtained, with the demand-side uncertainty analysis, for which data of comparable quality are obtainable only in hindsight.

Let C^* denote the value of C° which is obtained when the sizes of subsystems are chosen so that the user amount risk specifications ($\alpha_j \forall j$) are met, the cost risk specification (α_C) is met, and the value of C° is minimized.

Finally, the problem can be stated: find the technological alternatives that minimize C^* .





APPENDIX A

ALGORITHMIC DESCRIPTION OF THE SDTM COMPUTER PROGRAM

The letters i, j, k, t and w are used as index variables to represent subsystems, the resources they produce, cost types, time, and cost items respectively. Pseudocode is presented in boxes.

A.1 Core Analysis Control Logic

This section contains the high-level control logic for the core analysis algorithm. The following pieces of pseudocode describe what happens when the user runs an analysis.

A.1.1 The Analysis/Run Analysis Menu Pick

When the user selects the Analysis/Run Analysis pick, the following happens:

Compute resource sizes, halting on errors (Section A.4)

Compute all costs, halting on errors (Section A.5)

A.2 Load Spreaders, Profiles, etc.

SDTM time phasing runs from year T_{\min} to year T_{\max} ; the notation $\forall t$ means $\{t: T_{\min} \leq t \leq T_{\max}\}$. *BaseYear* is the base year for all real dollar amounts; *ThePresent* is the year to which costs are discounted; *AC* is the year in which assembly is complete; and *Life* is the lifetime of the station in years after *AC*. The following constraints on T_{\min} and T_{\max} exist:

$$\begin{aligned} T_{\min} &\leq AC \\ T_{\max} &= AC + Life - 1 \end{aligned}$$

G and K are the inflation and nominal discount rates in *BaseYear*; k is the real discount rate, which is assumed to be constant for all t . Note that

$$k = \left(\frac{K-G}{1+G} \right)$$

The real discount rate should be greater than zero; if an attempt is made to enter K such that $k < 0$, a warning will be generated.

The discounter for dollars in year t is

$$\text{disc}(t) = (1+k)^{\text{ThePresent}-t}$$

Define the present value and levelization of stream s_t as follows:

$$\text{pv}\{s_t\} = \sum_{t=T_{\min}}^{T_{\max}} s_t \times \text{disc}(t)$$

$$\text{lev}\{s_t\} = \text{pv}\{s_t\} \times \text{acrf}$$

The capital recovery factor, adjusted so that it is expressed as of *ThePresent* (rather than as of the start of the stream being considered, here *AC*, the date at which Space Station Freedom initial assembly is completed), is denoted *acrf*:

$$\text{acrf} = \frac{1}{\sum_{t=AC}^{T_{\max}} \text{disc}(t)}$$

The cost spreader for cost type k in year t is Q_{kt} ; it is defined $\forall t$. In fact, T_{\min} is defined to be the earliest year t for which $Q_{kt} > 0$ for some k .

Also, f_{jt} is the production profile for resource j in year t ; it is defined from *AC* to T_{\max} .

All of these values are loaded or computed before the sizing and costing algorithms begin.

A.3 Time Indexing

The sizing algorithm is concerned with the period of time from *AC* to T_{\max} . The costing algorithm is concerned with the period of time from T_{\min} to T_{\max} – an interval that includes *AC*. In the implementation of these algorithms, much data (spreader, discounters, production profiles) must be stored as vectors indexed by time.

If we arbitrarily assign to AC the index 1, then the sizing algorithm's time loops can run conveniently from 1 to $Life$. The sizing-related time vectors, then, should be dimensioned 1 to $MAX_LIFETIME$, where $MAX_LIFETIME$ is a program constant.

The costing algorithm runs from T_{min} to T_{max} . If we assign T_{min} and T_{max} indices based on the assumption that $AC = 1$, then the cost-related time vectors have an index set consistent with the sizing-related time vectors, and will start earlier than AC , at $-MAX_LEADTIME$, where $MAX_LEADTIME$ is another program constant.

In the C programming language, which is used for SDTM, vectors of size N are normally dimensioned 0 to $N-1$, but this can be changed. Sizing-related time vectors normally have size $MAX_LIFETIME + 1$, and are dimensioned 0 to $MAX_LIFETIME$. Costing-related time vectors have size $MAX_LEADTIME + MAX_LIFETIME + 1$, and are dimensioned from $-MAX_LEADTIME$ to $MAX_LIFETIME$, where index 1 still corresponds to AC .

A.4 The Sizing Algorithm

The sizing algorithm computes nameplate sizes consistent with the specified goal requirements, and also computes a variety of size-related results. This section contains pseudocode for the sizing algorithm.

The actual amount of a resource produced in a particular year is modeled as

$$X_{jt} \triangleq N_j f_{jt}$$

In the implementation of the algorithm, X_{jt} must be computed explicitly when N_j is changed. The pseudocode indicates where these computations must be done. Further, SDTM expressions can depend on the nameplate size, the actual size, and several functions of each, which must always be evaluated immediately so that they can be used in input formulas.

The overall flow of the sizing algorithm is as follows:

Initialize (see Section A.4.1)

Compute Nameplate Sizes (see Section A.4.2)

Compute the H matrix (see Section A.4.3)

A.4.1 Initialization

The following procedure initializes the sizing algorithm.

Evaluate all parameters

Initialize N_j by

$$N_j \leftarrow \begin{cases} \text{input reference size, } rN_j, & \text{if } GOAL_j = \text{USER,} \\ \text{input nameplate size specification, } GN_j, & \text{if } GOAL_j = \text{SIZE} \end{cases}$$

Compute $pv\{f_{jt}\} \forall j$ by

$$pv\{f_{jt}\} \leftarrow \sum_{t=AC}^{T_{\max}} f_{jt} \times \text{disc}(t)$$

Compute $\epsilon_j \leftarrow \max(rN_j, 1) \times \epsilon \forall j$

Compute $\delta_j \leftarrow \max(rN_j, 1) \times \delta \forall j$

The relatively small number ϵ_j is used in floating point comparisons involving resource j ; the virtual step δ_j is used in numerical differentiations involving N_j . The pure numbers ϵ and δ are global values and are controllable by input.

A.4.2 Computation of Nameplate Sizes

Given the nameplate size and user amount requirements for all resources, the goal specifications, and the consumption functions, the purpose of the sizing algorithm is to find the nameplate sizes that meet the specified goals, as well as the actual year-by-year sizes implied by the nameplate sizes and the profiles.

Loop until convergence (Gauss-Seidel):

Loop over $j \forall j$:

Compute $pv\{Y_{jt}\}$ using the latest sizes (this is the Seidel modification to Gauss's algorithm)

Evaluate the Goal user amount and size formulas for GU_j, GN_j

Compute $pv\{GU_j\} \leftarrow GU_j/acr_f$

Update N_j by

$$N_j \leftarrow \begin{cases} \frac{pv\{GU_j\} + pv\{Y_{jt}\}}{pv\{f_{jt}\}} & \text{if } GOAL_j = \text{USER} \\ GN_j & \text{if } GOAL_j = \text{SIZE} \end{cases}$$

Test for convergence, nonconvergence, and divergence (Section A.4.4)

A.4.3 Computing the H Matrix

Elements in the H matrix are defined as

$$H_{ji} \triangleq \frac{pv\left\{\frac{\partial A_{jil}}{\partial N_i}\right\}}{pv\{f_{jt}\}} \quad \forall j, i$$

where j is a resource and i is a resource-producing subsystem.

Compute H_{ji} as follows:

Loop over $i \forall i$:

Loop over $j \forall j$:

$pvDelta \leftarrow 0$

Loop over t from AC to T_{\max} :

Update $pvDelta$ by

$$pvDelta \leftarrow pvDelta + \frac{A_{jil}(N_i + \delta_i) - A_{jil}(N_i)}{\delta_i} \times \text{disc}(t)$$

$H_{ji} \leftarrow pvDelta/pv\{f_{jt}\}$

A.4.4 Convergence Criteria

The sizing algorithm uses the following tests for convergence, nonconvergence, and divergence.

A.4.4.1 Convergence

During each iteration of the Gauss-Seidel algorithm, the resource sizes are updated. The algorithm is judged to have converged when the condition

$$|N_j - N'_j| < \epsilon_j \quad \forall j$$

is true after two consecutive iterations, where N'_j denotes the value of N_j before it was updated. Two consecutive iterations must pass the test if three or more subsystems are being sized, because nonlinearities in the consumption functions, coupled with a Seidel change during an iteration, could conceivably move a computed size slightly away from optimality.

A.4.4.2 Nonconvergence

Pathological cases in which the Gauss-Seidel algorithm neither converges nor diverges explosively can be constructed (with difficulty, if realistic data is used) or might result from data entry errors. Any case in which the algorithm has neither converged nor diverged by a specified maximum number of iterations is judged to be nonconvergent.

When a nonconvergent case is found, the algorithm will halt and display an error message. No reports will be generated.

A.4.4.3 Divergence

If an infeasible station design is entered, the algorithm may diverge explosively; this situation, while rare, must be guarded against, as it could cause an arithmetic overflow that would halt the program. The test for this is quite simple and unsophisticated: the algorithm is judged to be diverging if

$$\exists j \ni \epsilon \times N_j > \max(rN_j, 1)$$

As above, ϵ is a small number; in SDTM, it is generally set to 10^{-7} .

In words, the algorithm is judged to be diverging if the nameplate size of a subsystem ever gets to be 10^7 times as large as its reference size. We use the larger of the reference size, rN_j , and 1 in case rN_j should happen to be zero (as might be the case for some subsystems).

A.5 The Costing Algorithm

The costing algorithm computes the life-cycle cost by cost item and resource based on the nameplate sizes computed by the sizing algorithm. The algorithm is based on the fundamental cost relationships described in Section A.5.1.

The overall flow of the costing algorithm is as follows:

Initialize (see Section A.5.2)

Compute Total Life-Cycle Cost (see Section A.5.3)

Save LCC and LCC_k as LCC' and LCC'_k

Compute Explicit Resource Costs (see Section A.5.4)

Compute Implicit Resource Costs and Revenues (see Section A.5.5)

Re-compute Total Life-Cycle Cost (see Section A.5.3)

Computing the total life-cycle cost involves computing all of the costs at every level of the cost tree; the life-cycle costs are simply the costs at the root of the tree. The life-cycle cost must be computed repeatedly during the explicit cost calculations for different resource sizes; although the final LCC and LCC_k are saved after the first computation, the costs throughout the rest of the tree must be recalculated. Thus, the final step of the overall algorithm is to compute LCC one final time.

A.5.1 Fundamental Relationships

The value of cost item w 's k th cost formula is E_{wk} , the add-on cost of cost item w and cost type k . The wrap fraction for cost item w and cost type k is W_{wk} . The spreader for cost type k in year t is Q_{kt} . The fundamental cost relationships are as follows, where Ch is a child of item w :

$$CT_{wk} = E_{wk} \times pv\{Q_{kt}\} + \sum_{Ch} (1 + W_{Ch,k}) CT_{Ch,k}$$

$$C_{wkt} = \frac{CT_{wk} Q_{kt}}{pv\{Q_{kt}\}}$$

$$CKT_w = \sum_k CT_{wk}$$

$$LCC = CKT_{root}$$

$$LCC_k = CT_{root,k}$$

Note that if spreaders were allowed to vary by cost item, the fundamental equation would become

$$C_{wkt} = E_{wk} Q_{kt} + \sum_{Ch} (1 + W_{Ch,k}) C_{Ch,k,t}$$

This formulation, while allowing more flexibility in the spreaders, would also use much more memory space. Additionally, one could allow E_{wk} to depend on the X_{ji} as well as on the N_j .

A.5.2 Costing Initialization

The only initialization necessary for the cost algorithm is the computation of $pv\{Q_{kt}\}$.

A.5.3 Computing the Life-Cycle Cost

This is the algorithm used to compute the life-cycle cost in accordance with the fundamental relationships given above. This algorithm is used repeatedly to compute not only the LCC , but also the explicit and implicit costs for each resource.

There are N_w cost items. Order them by level in the cost tree, so that the first cost item is the root of the tree and the N_w th cost item is a cost item on the bottom-most level. Number them $1, \dots, N_w$. Let w' denote the number of the parent of item w . (The parent of item 1 is undefined.) Then, use the following algorithm:

```

Initialize  $CT_{wk} \leftarrow 0 \quad \forall w, k.$ 
Initialize  $CKT_w \leftarrow 0 \quad \forall w.$ 
Loop over cost items  $w$  from  $N_w$  down to 1:
    Loop over cost types  $k \quad \forall k:$ 
        Evaluate  $E_{wk}, W_{wk}.$ 
         $CT_{wk} \leftarrow CT_{wk} + E_{wk}.$ 
         $CKT_w \leftarrow CKT_w + CT_{wk}.$ 
        If  $w$  is not the root (i.e.,  $w \neq 1$ ),
             $CT_{w'k} \leftarrow CT_{w'k} + (1 + W_{wk})CT_{wk}.$ 

```

A.5.4 Computing the Explicit Resource Costs

The life-cycle cost, $LCC \triangleq CKT_{root}$, is a function of N_i for all i . The explicit cost of subsystem i is a function of N_i alone. The explicit cost is defined as follows:

$$C_{ik}(N_i) \triangleq LCC_k(N_j \ \forall j) - LCC_k(N_{j \neq i}, N_i = 0)$$

$$ExplCost_i \triangleq \sum_k C_{ik}$$

The pseudocode for $ExplCost_i$ is as follows:

Loop over resource $i \ \forall i$:

Save N_i

$N_i \leftarrow 0$

$ExplCost_i \leftarrow 0$

Compute $LCC_k \ \forall k$ (see Section A.5.3)

Loop over cost type $k \ \forall k$:

$C_{ik} \leftarrow LCC'_k - LCC_k$

$ExplCost_i \leftarrow ExplCost_i + C_{ik}$

Restore N_i

There may be some costs that are not allocated to any subsystem. This unallocated cost is calculated by

$$\text{Unallocated } LCC \triangleq LCC - \sum_i ExplCost_i$$

Now, for use in the next subsection, compute $SLCC_j \ \forall j$. $SLCC_j$ is the slope of the life-cycle cost with respect to the nameplate size of subsystem j . $SLCC_j$ is defined as follows:

$$SLCC_j \triangleq \frac{\partial LCC}{\partial N_j}$$

$$\approx \frac{LCC(N_j + \delta_j, N_{i \neq j}) - LCC(N_i \ \forall i)}{\delta_j}$$

where δ_j is a virtual change in the nameplate size of resource j . This is the pseudocode for $SLCC_j$:

```

Loop over resource  $j \quad \forall j$ :
    Save  $N_j$ 
     $N_j \leftarrow N_j + \delta_j$ 
    Compute  $LCC$  (see Section A.5.3)
     $SLCC_j \leftarrow (LCC - LCC')/\delta_j$ 
    Restore  $N_j$ 

```

A.5.5 Computing the Implicit Resource Costs and Revenues

The marginal cost of an additional user requirement for resource i is denoted MCU_i , and is the total amount the station life-cycle cost would increase if one more unit of resource i were made available to station users (or used by the station itself) in each year of the station's life. The levelized marginal cost is denoted LEV_i , and can be thought of as the amount a user might pay for one more unit of resource i in a particular year if a marginal-cost-recovering pricing policy were being used. The implicit cost of resource i is denoted $ImplCost_i$. The implicit revenue is denoted $ImplRevenue_i$.

The marginal cost, MCU_i , depends on $SLCC_i$ as defined above, and on the H matrix (Section A.4.3). MCU_i and LEV_i are calculated as follows:

```

Initialize  $LEV_i \leftarrow SLCC_i / pv\{f_{it}\} \quad \forall i$  (producers)
Loop until convergence (Gauss-Seidel):
    Loop over  $i \quad \forall i$  (producers):
        Initialize Sum  $\leftarrow 0$ 
        Loop over  $j \quad \forall j$  (producers):
            Sum  $\leftarrow$  Sum +  $LEV_j H_{ji}$ 
         $LEV_i \leftarrow$  Sum +  $SLCC_i / pv\{f_{it}\}$ 
    Test for convergence, nonconvergence, and divergence (Section A.5.6)
If  $LEV_i < 0$  for any  $i$ , generate an error message and halt
Calculate lifetime  $MCUs$  by
     $MCU_i \leftarrow LEV_i / acrf \quad \forall i$ 

```


Finally, $ImplCost_i$ is computed as follows:

```

Loop over  $i \ \forall i$  (consumers):
    Initialize  $ImplCost_i \leftarrow 0$ 
    Loop over  $j \ \forall j$  (producers):
         $ImplCost_i \leftarrow ImplCost_i + MCU_j \times lev\{A_{ji}\}$ 
     $ImplRevenue_i \leftarrow MCU_i \times lev\{X_{ii}\}$ 

```

A.5.6 Marginal Cost Convergence Criteria

The marginal cost computation algorithm uses a Gauss-Seidel loop, which uses the following tests for convergence, nonconvergence, and divergence.

A.5.6.1 Convergence

During each iteration of the Gauss-Seidel algorithm, the LEV_i are updated. The algorithm is judged to have converged when

$$|LEV_i - LEV'_i| < \epsilon \times \left| \frac{SLCC_i}{pv(f_{ii})} \right| \quad \forall i$$

is true after two consecutive iterations. LEV'_i denotes the value of LEV_i before it was updated.

A.5.6.2 Nonconvergence

Pathological cases in which the Gauss-Seidel algorithm neither converges nor diverges explosively can be constructed or might result from data entry errors. Any case in which the algorithm has neither converged nor diverged by the specified maximum number of iterations is judged to be nonconvergent.

When a nonconvergent case is found, the algorithm will halt and display an error message. No reports will be generated.

A.5.6.3 Divergence

If an infeasible station design is entered, the algorithm may diverge explosively. The test for this is quite simple and unsophisticated: the algorithm is judged to be diverging if

$$\exists i \ni \epsilon \times |LEV_i| > \max \left(\left| \frac{SLCC_i}{pv\{f_{it}\}} \right|, 1 \right)$$

As above, ϵ is a small number; in SDTM, it is generally set to 10^{-7} .

APPENDIX B

THE EXPRESSION OF LEVELIZED AND LIFE-CYCLE COSTS

Suppose it is estimated that it will cost 3000 constant 1990 dollars per pound to deliver payloads to orbit in year 2000. At 5 percent per year inflation, \$4887 per pound would actually have to be spent. With a 10 percent per year discount rate, only \$1884 would have to be invested in 1990 to pay for each pound to be lifted in 2000. All of these numbers are "correct"; the issue is how to express the cost in the least misleading way.

Inflation describes how the measuring stick for trade value changes with time. Escalation, which could be (but seldom is) called "differential inflation," describes changes in the relative trade values of particular goods and services. Cost estimates or prices stated in terms of the amounts of money that would actually change hands are said by economists to be expressed in "nominal" dollars. The phrase "real-year" dollars is sometimes used for this concept.

Cost estimates are often stated in "constant" dollars associated with a specified base year. The base year used is often the year the estimate was made, because no adjustment then has to be made for inflation or escalation. "Constant <base year> dollars" are sometimes called "real dollars" by economists because money has extrinsic value only in terms of trades of goods and services, and that value does not change with general inflation (though it does change with escalation).

Because resources can be used, with time, to make more resources, the timing of expenditures or receipts also affects their value. Different cost and/or revenue streams can be meaningfully compared in terms of the amounts of money that would have to be invested at some "present" – provided, of course, that the alternatives are associated with identical streams of goods and services produced. Usually, the major implication of this qualification is that all alternatives whose costs are being compared must have the same lifetime.

The restriction that the streams of goods and services provided by design alternatives must be identical can be restated in terms of *levelization*. The most commonly encountered form of levelization is a stream of payments that are constant in nominal terms for a specified period, as in a conventional home mortgage. A slightly more sophisticated version is payments that are constant "in real terms"; that is, that grow at the rate of inflation. Both of these forms of levelization implicitly assume that what is being paid for by the levelized payment is either a pure financial transaction (as in paying off a loan or in buying an annuity) or is to be delivered at a constant rate. In the latter case, real levelization obviates the need to assume equal lifetimes among alternatives.

If the product is not delivered at a constant rate, a slightly more general form of levelization, in which the same number of real dollars is associated with each unit of the goods or service delivered, is needed. The number of dollars to assign to each unit is that which would equate the present value of the "revenue" stream to the present value of the "cost" stream.

(Quotation marks are used with "revenue" and "cost" because it is not necessary that any funds actually change hands: the cash-flow streams can be strictly hypothetical.)

It is quite correct, but also quite misleading, to assert that \$1884 *now* is equivalent to 3000 constant 1990 dollars in 2000. It is reasonable to assume that no customer for station resources is going to make an investment now to pay for costs to be incurred later. A more useful assumption is that station customers will "pay for" resources as they are used.

Subsystems, on the other hand, do not "buy" on-board resources by the unit. Instead, their demand is for a fraction of the capacity for the lifetime of the station. Furthermore, their design tradeoff decisions do occur *now*, not at the time the resources are used.

Consequently, the unlevelized marginal costs should be expressed in the same terms as the life-cycle cost, but per capacity unit.

Both levelized and unlevelized costs are defined in terms of the life-cycle cost:

MCU_j = (Unlevelized) marginal cost of resource j : the amount by which the life-cycle cost would increase if the amount of resource j made available to station users were held constant but the housekeeping requirement increased, per unit of increase in housekeeping demand.

LEV_j = Levelized marginal cost of resource j : the amount by which the life-cycle cost would increase if the user amount of resource j were increased, expressed in undiscounted constant base year dollars per unit of net supply.

As stated in Appendix A, the levelized and unlevelized costs are related through the adjusted capital recovery factor.

$$LEV_j = MCU_j \cdot acrf(k, L)$$

where

k = the real discount rate.

L = station lifetime.